



UNIVERSITI PUTRA MALAYSIA

**A FASTER VERSION OF RIJNDAEL CRYPTOGRAPHIC
ALGORITHM USING CYCLIC SHIFT AND
BITWISE OPERATIONS**

FAKARIAH HANI BT MOHD ALI

FSKTM 2004 5

**A FASTER VERSION OF RIJNDAEL CRYPTOGRAPHIC
ALGORITHM USING CYCLIC SHIFT AND
BITWISE OPERATIONS**

By

FAKARIAH HANI BT MOHD ALI

**Thesis Submitted to the School of Graduate Studies, Universiti Putra
Malaysia, in Fulfilment of the Requirements for the
Degree of Master of Science**

February 2004



*Dedicated to my husband; Mohd Rafaie Abdul Razak,
my daughter; Nurul Rusydina, my newborn baby; Muhammad Fawwaz Hadi
my parents and family*

Abstract of thesis presented to the Senate of Universiti Putra Malaysia in
fulfilment of the requirements for the degree of Master of Science

**A FASTER VERSION OF RIJNDAEL CRYPTOGRAPHIC
ALGORITHM USING CYCLIC SHIFT AND
BITWISE OPERATIONS**

By

FAKARIAH HANI BT MOHD ALI

February 2004

Chairman: Associate Professor Ramlan Mahmod, Ph.D.

Faculty: Computer Science and Information Technology

Doing arithmetic in finite field is the key part to the implementation of communication and coding system including the newly developed Rijndael the Advanced Encryption Standard (AES). This encryption standard uses KeyExpansion, ByteSub, Mixcolumn and Shiftrow functions which consists of XOR, inverse, multiplying and swap modules. Among them, inverse and multiplier are the most complex modules with longer delay. These modules are included in the Mixcolumn function. From the proposal of AES, the Mixcolumn function was suggested to solve the problem of delay by using look-up tables. This function can be integrated into a bigger table to replace the calculations of inverse and multiply operations, if it provides enough memory. In fact, too many tables are needed for various irreducible polynomials that this system is not flexible and expandable. The area for lookup tables becomes huge

when multiple round units are implemented. This research proposes the use of cyclic shift and bit wise XOR operation as new approach to replace the lookup table. The principle benefit of using this new approach over the transform from Rijndael block cipher is speed. This new approach has shown the excellent result, which faster then Rijndael. The new approach algorithm speed increment has consistently increased in between 18% to 22% microsecond for encryption and 30% to 34% for decryption compared to Rijndael algorithm.

Abstrak tesis yang dikemukakan kepada Senat Universiti Putra Malaysia
sebagai memenuhi keperluan untuk ijazah Master Sains

**VERSI TERLAJU ALGORITMA
KRIPTOGRAFI RIJNDAEL DENGAN MENGGUNAKAN
OPERASI SHIFT DAN BITWISE**

Oleh

FAKARIAH HANI MOHD ALI

Februari 2004

Pengerusi: Profesor Madya Ramlan Mahmod, Ph.D.

Fakulti: Sains Komputer dan Teknologi Maklumat

Penggunaan aritmetik dalam operasi nombor yang terbatas (finite field) adalah kunci kepada pelaksanaan sistem komunikasi dan pengkodan. Ia juga digunapakai oleh Rijndael yang dikenali sebagai Advanced Encryption Standard. Algoritma ini melaksanakan fungsi KeyExpansion, ByteSub, Mixcolumn dan Shiftrow yang mengandungi operasi XOR, songsangan, pendaraban dan pertukaran. Dalam kesemua operasi, operasi songsangan dan pendaraban adalah operasi yang dikenalpasti boleh menyebabkan kelewatan. Operasi ini dilaksanakan dalam fungsi Mixcolumn. Daripada kertas cadangan AES, fungsi Mixcolumn telah dicadangkan untuk mengatasi masalah kelewatan ini dengan menggunakan *jadual*. Fungsi ini diintegrasikan ke dalam satu *jadual* yang besar untuk menggantikan operasi songsangan dan pendaraban sekiranya cukup memori. Lagipun terlalu banyak jadual diperlukan untuk berbagai “irreducible polynomials” dan menyebabkan ianya tidak sesuai dan

tidak boleh dikembangkan. Jadual juga akan melibatkan penggunaan ruang yang amat besar apabila dilaksanakan pada banyak pusingan . Penyelidikan ini mencadangkan untuk menggunakan operasi *shift* dan *XOR* untuk menggantikan penggunaan jadual tersebut. Kelebihan penggunaan pendekatan baru ini ialah pertambahan dari sudut kelajuan. Pendekatan ini telah menunjukkan keputusan yang agak memberansangkan apabila ia di adaptasikan kepada algoritma Rijndael. Keseluruhannya pendekatan baru ini telah menyebabkan kelajuan algoritma Rijndael bertambah secara konsisten sebanyak diantara 18% hingga 22% mikrosaat untuk proses enkripsi dan 30% hingga 34% untuk proses dekripsi.

ACKNOWLEDGEMENTS

Praise to ALLAH S.W.T for giving me strength, patience, and motivation to complete this research work.

My deepest appreciation and gratitude go to the research committee lead by *Prof.Madya Dr. Ramlan Mahmud, Dr. Muhammad Rushdan Mohd. Said* and *Dr Ismail Abdullah* for providing me inspiration for this work and also for their virtuous guidance, encouragement, support and help during the time of doing the research.

My deepest thanks go to my husband, parents, family, and colleagues at the Faculty of Computer Science and Information Technology, UPM, for their supports and encouragement during the time of doing the research.

For financial support, I was grateful to Universiti Teknologi MARA for giving me the scholarship while I was carrying out this research. Thank you all and may God bless all these individuals for their kindness.

Fakariah Hani Mohd Ali

February 2004



I certify that an Examination Committee met on 17th February 2004 to conduct the final examination of Fakariah Hani Mohd Ali on her Master of Science thesis entitled “A Faster Version of Rijndael Cryptographic Algorithm Using Cyclic Shift and Bitwise Operations” in accordance with Universiti Pertanian Malaysia (Higher Degree) Act 1980 and Universiti Pertanian Malaysia (Higher Degree) Regulations 1981. The Committee recommends that the candidate be awarded the relevant degree. Members of the Examination Committee are as follows:

Hj. Mohd. Hassan Selamat, Ph.D.

Associate Professor
Faculty of Computer Science and information Technology
Universiti Putra Malaysia
(Chairman)

Hj. Ali Mamat, Ph.D.

Associate Professor
Faculty of Computer Science and information Technology
Universiti Putra Malaysia
(Member)

Abdul Azim Abd Ghani, Ph.D.

Associate Professor
Faculty of Computer Science and information Technology
Universiti Putra Malaysia
(Member)

Abdullah Md. Zin, Ph.D.

Associate Professor
Faculty of Technology and Information Science
Universiti Kebangsaan Malaysia
(Independent Examiner)



GULAM RUSUL RAHMAT ALI, Ph.D.
Professor/Deputy Dean
School of Graduate Studies
Universiti Putra Malaysia

Date: 07 JUN 2004

This thesis submitted to the Senate of Universiti Putra Malaysia and has been accepted as fulfilment of the requirements for the degree of Master of Science. The members of the Supervisory Committee are as follows:

Ramlan Mahmod, Ph.D.

Associate Professor
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Member)

Muhammad Rushdan Mohd Said, Ph.D.

Associate Professor
Faculty of Science and Environmental Studies
Universiti Putra Malaysia
(Member)

Ismail Abdullah, Ph.D.

Associate Professor
Institute for Mathematical Research
Universiti Putra Malaysia
(Member)



AINI IDERIS, Ph.D.

Professor/Dean
School of Graduate Studies
Universiti Putra Malaysia

Date : 16 JUN 2004

DECLARATION

I hereby declare that the thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at UPM or other institutions.


(FAKARIAH HANI MOHD ALI)

Date : 10 June 2004

TABLE OF CONTENTS

| | Page |
|--------------------------------------|----------------|
| DEDICATION | ii |
| ABSTRACT | iii |
| ABSTRAK | v |
| ACKNOWLEDGEMENTS | vii |
| APPROVAL | viii |
| DECLARATION | x |
| LIST OF TABLES | xiii |
| LIST OF FIGURES | xv |
| LIST OF ABBREVIATIONS | xvi |
| CHAPTER | |
| 1 INTRODUCTION | 1.1 |
| 1.1 Background | 1.4 |
| 1.2 Problem Statement | 1.5 |
| 1.3 Scope of research | 1.5 |
| 1.4 Objective | 1.5 |
| 1.5 Thesis Overview | 1.5 |
| 2 LITERATURE REVIEW | 2.1 |
| 2.1 The Basics of Cryptography | 2.1 |
| 2.2 Encryption and Decryption | 2.1 |
| 2.3 What is Cryptography? | 2.2 |
| 2.3.1 How does Cryptography work? | 2.3 |
| 2.4 Cryptanalysis | 2.3 |
| 2.5 Public Key Cryptosystems | 2.6 |
| 2.6 Conventional Cryptosystems | 2.9 |
| 2.6.1 Data Encryption Standard (DES) | 2.11 |
| 2.7 New Symmetric key Algorithm | 2.13 |
| 2.7.1 Rijndael | 2.14 |
| 2.7.2 Twofish | 2.15 |
| 2.7.3 Serpent | 2.15 |
| 2.7.4 RC6 TM | 2.16 |
| 2.7.5 MARS | 2.16 |
| 2.8 Advanced Encryption Standard | 2.17 |
| 2.8.1 AES Overview | 2.17 |
| 2.9 AES and DES Comparisons | 2.18 |
| 2.10 AES related research | 2.20 |
| 2.11 Summary | 2.21 |



3 RIJNDAEL THE AES

| | |
|--|------|
| 3.1 Introduction | 3.1 |
| 3.2 Definitions | 3.2 |
| 3.2.1 Glossary of Terms and Acronyms | 3.2 |
| 3.2.2 Algorithm Parameters, Symbols and Functions | 3.3 |
| 3.3 Notation and Conventions | 3.5 |
| 3.3.1 Inputs and Outputs | 3.5 |
| 3.3.2 Bytes | 3.6 |
| 3.3.3 The State | 3.7 |
| 3.4 Mathematical Preliminaries | 3.8 |
| 3.4.1 Addition | 3.8 |
| 3.4.2 Multiplication | 3.9 |
| 3.4.3 Finite field Multiplication Using Tables | 3.10 |
| 3.4.4 Polynomials with Coefficients in $GF(2^8)$ | 3.12 |
| 3.5 Algorithm Specification | 3.15 |
| 3.5.1 Cipher | 3.16 |
| 3.5.2 Key Expansion | 3.24 |
| 3.5.3 Inverse Cipher | 3.26 |
| 3.5.4 Inverse of the AddRoundkey () transformation | 3.31 |

4 METHODOLOGY

| | |
|--|------|
| 4.1 Introduction | 4.1 |
| 4.2 Methodology | 4.1 |
| 4.2.1 Input | 4.3 |
| 4.2.2 Experiment | 4.4 |
| 4.3 Method and Technique used | 4.6 |
| 4.3.1 Mixcolumns () transformation | 4.7 |
| 4.3.2 InvMixcolumns () transformation | 4.8 |
| 4.3.3 Rijndael Finite field Multiplication Using Lookup Tables | 4.10 |
| 4.4 Cyclic Shift and XOR operation | 4.13 |
| 4.4.1 Cyclic Shift | 4.13 |
| 4.4.2 Combination of Shift and Bit Wise Exclusive OR (XOR) | 4.15 |

5 RESULTS AND DISCUSSION

| | |
|--|------|
| 5.1 Performance Comparison Between Rijndael and New Approach | 5.1 |
| 5.1.1 Simple plaintexts | 5.2 |
| 5.1.2 Half paragraph of plaintexts | 5.11 |
| 5.1.3 A paragraph of plaintexts | 5.19 |
| 5.1.4 Two paragraphs of plaintext | 5.28 |
| 5.1.5 One page of plaintexts | 5.37 |
| 5.2 Input output of the encryption and decryption process | 5.46 |
| 5.2.1 Simple plaintext | 5.46 |
| 5.2.2 Half paragraph of plaintexts | 5.49 |

| | |
|-------------------------------------|------------|
| 6 CONCLUSION AND FUTURE WORK | 6.1 |
| 6.1 Conclusion | 6.1 |
| 6.2 Suggestion for future work | 6.3 |
| REFERENCES | R.1 |
| BIODATA OF AUTHOR | B.1 |

LIST OF TABLES

| Tables | Page |
|---|------|
| 2.1 Types of Attack on encrypted Message | 2.4 |
| 2.2 The functional similarities between the AES and DES | 2.18 |
| 2.3 The differences between the AES and DES | 2.19 |
| 3.1 'Logs' – L values such that $\{xy\} = \{03\}^L$ for a given finite field element $\{xy\}$ | 3.11 |
| 3.2 'AntiLogs' – field element $\{E\}$ such that $\{E\} = \{03\}^{xy}$ given the power $\{xy\}$ | 3.11 |
| 4.1 Key, key length and block length used for both algorithms | 4.5 |
| 4.4 Shift left by one bit operation | 4.14 |
| 4.5 Finite field multiplication of '02' in binary and hexadecimal | 4.14 |
| 4.6 Finite field multiplication of '03' in binary and hexadecimal | 4.15 |
| 4.7 New approach for '03' | 4.16 |
| 4.8 Finite field multiplication of '0E' in binary and hexadecimal | 4.17 |
| 4.9 New approach for '0E' | 4.17 |
| 4.10 Finite field multiplication of '0D' in binary and hexadecimal | 4.18 |
| 4.11 New approach for '0D' | 4.18 |
| 4.12 Finite field multiplication of '0B' in binary and hexadecimal | 4.19 |
| 4.13 New approach for '0B' | 4.20 |
| 4.14 Finite field multiplication of '09' in binary and hexadecimal | 4.20 |
| 4.15 New approach for '09' | 4.21 |



| | | |
|------|---|------|
| 5.1 | Experiment have been studied for both algorithms | 5.2 |
| 5.2 | Simple plaintexts with time in microseconds during encryption process | 5.3 |
| 5.3 | Simple plaintexts with time in microseconds during decryption process | 5.7 |
| 5.4 | Half paragraph plaintexts with time in microseconds during encryption process | 5.12 |
| 5.5 | Half paragraph plaintexts with time in microseconds during decryption process | 5.15 |
| 5.6 | A paragraph of plaintext with time in microseconds during encryption process | 5.20 |
| 5.7 | A paragraph of plaintext with time in microseconds during decryption process | 5.24 |
| 5.8 | Two paragraphs of plaintexts with time in microseconds during encryption process | 5.29 |
| 5.9 | Two paragraphs of plaintexts with time in microseconds during decryption process | 5.33 |
| 5.10 | One page of plaintexts with time in microseconds during encryption process | 5.38 |
| 5.11 | One page of plaintexts with time in microseconds during decryption process | 5.42 |
| 5.12 | Input of the encryption process using Rijndael algorithm | 5.46 |
| 5.13 | Input of the encryption process using new approach algorithm | 5.48 |
| 5.14 | Input of the encryption process using Rijndael algorithm | 5.49 |
| 5.15 | Input of the encryption process using new approach algorithm | 5.52 |

LIST OF FIGURES

| Figure | Page |
|---|------|
| 1.0 A symmetric cryptosystem | 1.1 |
| 1.1 A Public key cryptosystem | 1.2 |
| 2.1 Encryption and decryption | 2.2 |
| 2.2 Public key encryption | 2.7 |
| 2.3 Conventional encryption | 2.10 |
| 2.4 Enciphering computation | 2.13 |
| 3.1 Key-block round combinations | 3.15 |
| 3.2 SubBytes() applies the S-Box to each byte of the state | 3.18 |
| 3.3 S-Box: Substitution values for the byte xy | 3.19 |
| 3.4 ShiftRows () cyclically shifts the last three rows in the state | 3.20 |
| 3.5 Mixcolumn () operates on the State column-by-column | 3.22 |
| 3.6 AddRoundkey () XORs each column of the state with a from the key schedule | 3.24 |
| 3.7 InvShiftrows () cyclically shifts the last three rows in the state | 3.28 |
| 3.8 Inverse S-Box: Substitution values for the byte xy | 3.29 |
| 5.1 Simple plaintext using 128 bits key length and 128, 192, 256 bits block length | 5.4 |
| 5.2 Simple plaintext using 192 bits key length and 128, 192, 256 bits block length | 5.5 |
| 5.3 Simple plaintext using 256 bits key length and 128, 192, 256 bits block length | 5.6 |
| 5.4 Simple plaintext using 128 bits key length and 128, 192, 256 bits block length | 5.8 |

| | | |
|------|---|------|
| 5.5 | Simple plaintext using 192 bits key length and 128, 192, 256 bits block length | 5.9 |
| 5.6 | Simple plaintext using 256 bits key length and 128, 192, 256 bits block length | 5.10 |
| 5.7 | A half paragraph of plaintext | 5.11 |
| 5.8 | Half paragraph of plaintext using 128 bits key length and 128, 192, 256 bits block length | 5.13 |
| 5.9 | Half paragraph of plaintext using 192 bits key length and 128, 192, 256 bits block length | 5.14 |
| 5.10 | Half paragraph of plaintext using 256 bits key length and 128, 192, 256 bits block length | 5.15 |
| 5.11 | Half paragraph of plaintext using 128 bits key length and 128, 192, 256 bits block length | 5.16 |
| 5.12 | Half paragraph of plaintext using 192 bits key length and 128, 192, 256 bits block length | 5.17 |
| 5.13 | Half paragraph of plaintext using 256 bits key length and 128, 192, 256 bits block length | 5.18 |
| 5.14 | A paragraph of plaintext | 5.19 |
| 5.15 | A paragraph of plaintext using 128 bits key length and 128, 192, 256 bits block length | 5.21 |
| 5.16 | A paragraph of plaintext using 192 bits key length and 128, 192, 256 bits block length | 5.22 |
| 5.17 | A paragraph of plaintext using 256 bits key length and 128, 192, 256 bits block length | 5.23 |
| 5.18 | A paragraph of plaintext using 128 bits key length and 128, 192, 256 bits block length | 5.25 |
| 5.19 | A paragraph of plaintext using 192 bits key length and 128, 192, 256 bits block length | 5.26 |
| 5.20 | A paragraph of plaintext using 256 bits key length and 128, 192, 256 bits block length | 5.27 |

| | | |
|------|--|------|
| 5.21 | Two paragraph of plaintext | 5.28 |
| 5.22 | Two paragraph of plaintext using 128 bits key length and 128, 192, 256 bits block length | 5.30 |
| 5.23 | Two paragraph of plaintext using 192 bits key length and 128, 192, 256 bits block length | 5.31 |
| 5.24 | Two paragraph of plaintext using 256 bits key length and 128, 192, 256 bits block length | 5.32 |
| 5.25 | Two paragraph of plaintext using 128 bits key length and 128, 192, 256 bits block length | 5.34 |
| 5.26 | Two paragraph of plaintext using 192 bits key length and 128, 192, 256 bits block length | 5.35 |
| 5.27 | Two paragraph of plaintext using 256 bits key length and 128, 192, 256 bits block length | 5.36 |
| 5.28 | One page of plaintext | 5.37 |
| 5.29 | One page of plaintext using 128 bits key length and 128, 192, 256 bits block length | 5.39 |
| 5.30 | One page of plaintext using 192 bits key length and 128, 192, 256 bits block length | 5.40 |
| 5.31 | One page of plaintext using 256 bits key length and 128, 192, 256 bits block length | 5.41 |
| 5.32 | One page of plaintext using 128 bits key length and 128, 192, 256 bits block length | 5.43 |
| 5.33 | One page of plaintext using 192 bits key length and 128, 192, 256 bits block length | 5.44 |
| 5.34 | One page of plaintext using 256 bits key length and 128, 192, 256 bits block length | 5.45 |
| 5.35 | The output of the encryption process | 5.46 |
| 5.36 | Input of decryption process using Rijndael | 5.47 |
| 5.37 | Output of the decryption process | 5.47 |

| | | |
|------|--------------------------------------|------|
| 5.38 | The output of the encryption process | 5.48 |
| 5.39 | Input of decryption process | 5.48 |
| 5.40 | Output of the decryption process | 5.49 |
| 5.41 | The output of the encryption process | 5.50 |
| 5.42 | Input of decryption process | 5.51 |
| 5.43 | Output of the decryption process | 5.43 |
| 5.44 | Output of the encryption process | 5.52 |
| 5.45 | The input of decryption process | 5.53 |
| 5.46 | The output of the decryption process | 5.53 |



LIST OF ABBREVIATIONS

| | | |
|------|---|--|
| NIST | - | National Institute of Standards and Technology |
| AES | - | Advanced Encryption Standard |
| DES | - | Data Encryption Standard |
| GF | - | Galois Field |
| XOR | - | Exclusive XOR |
| ATM | - | Automated Teller Machine |
| NBS | - | National Bureau of Standard |
| DEA | - | Data Encryption Algorithm |
| RSA | - | Rivest Shamir Adlemen |
| VCR | - | Video cassette recorders |
| ISO | - | International Organization for Standardization |
| IEEE | - | Institute of Electric Electronic Engineering |
| ANSI | - | American National Standards Institute |
| IETF | - | Internet Engineering Task Force |

CHAPTER 1

INTRODUCTION

1.1 Background

Cryptography is the science of using mathematics to encrypt and decrypt data. Cryptography enables us to store sensitive information or transmit it across insecure networks (like the Internet) so that it cannot be read by anyone except the intended recipient. Cryptography is the science of secret writing. It has been used since the dawn of writing itself to conceal messages from an adversary. Cryptography distinguishes between two types of cryptographic system: symmetric and public key. Both encrypt messages using computer algorithm, and both provide users with secrecy through the use of cryptographic keys. The difference between the two cryptosystems lies in the way keys are used. A symmetric cryptosystem has a single key (see Figure 1), which is used to both encrypt and decrypt messages. The algorithm is a mathematical process that transforms plaintext into ciphertext and back again, and each transformation depends on the value of the key.

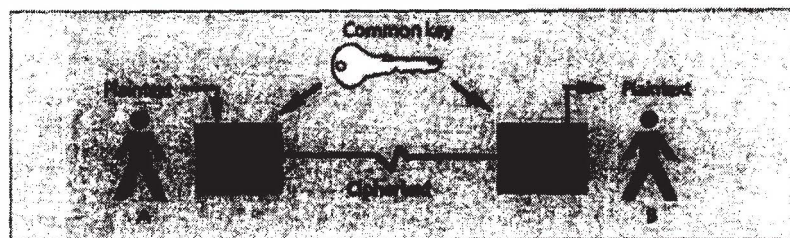


Figure 1: A symmetric cryptosystem

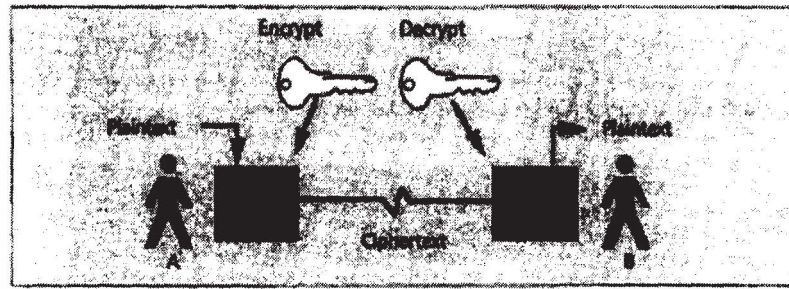


Figure 1.1: A public key cryptosystem

A well-known example of a symmetric cryptosystem is Data Encryption Standard (Federal Information Processing Standards Publication (FIPS PUB) 46,1977) (Coppersmith et. al., 1997). Other schemes include Triple DES, IDEA, RC4, RC5 and Blowfish. The new technology known as public key cryptography was introduced in 1976. In contrast to symmetric cryptosystems, public-key cryptosystems use complementary pairs of keys to separate the functions of encryption and decryptions as shown in Figure 1.1. One key is in a pair, and the private key is kept secret but the other is made publicly. Detail description for both cryptosystems will be explained in chapter 2. This research focused on symmetric cryptosystem only.

Improvement in the speed and power of microprocessor chips has made the Data Encryption Standard with its 56-bit key becomes subject to brute-force attack that can be carried out by organizations of moderate size. In 1997, the National Institute of Standards and Technology (NIST) US (an agency of the U.S Department of Commerce's Technology Administration) initiated a process to select a symmetric-key encryption algorithm to be used to protect sensitive (unclassified) Federal information in furtherance of NIST's statutory responsibilities and to be adopted as Advanced Encryption Standard

(AES). This new algorithm will replace Data Encryption Standard, which has been standard since 1977. In 1998, NIST announced the acceptance of fifteen candidate algorithms and requested the assistance of the cryptographic research community in analyzing the candidates. NIST reviewed the results of this preliminary research and selected MARS, RC6™, Rijndael, Serpent and Twofish as finalists. Having reviewed further public analysis of the finalists, NIST has decided to propose Rijndael (Daemen et. al., 1999) as the Advanced Encryption Standard (AES) (Nechvatal et. al., 2000) (AES Development Effort, 2001).

Rijndael is a block cipher designed by Joan Daemen and Vincent Rijmen. According to Nechvatal et. al. (2000), Rijndael's combination of security, performance, efficiency, implementability and flexibility makes it an appropriate selection for the AES for use in the technology of today and in the future. Detail description on Rijndael will be explained in chapter 3. There are many further analyses and improvements have been done on Rijndael (Daemen et. al., 1999) (Federal Information Processing Standards Publication (FIPS) 197,2001). McLoone et. al. (2001) proposed a Field Programmable Gate Arrays (FPGAs) Rijndael encryption design which utilizes look-up tables to implement the entire Rijndael Round function. Jing et. al. (2001) proposed a new algorithm for computing inverse in $GF(2^m)$ on the standard basis. They proposed a set of multiplier and inverse in $GF(2^m)$ to increase the computing speed. Sklavos et. al. (2002) proposed an alternative architectures and VLSI implementation designs. These designs operate for both encryption and decryption process in the same device. Xinmiao et. al.

(2002) addresses various approaches for efficient hardware implementation of the AES algorithm.

1.2 Problem Statement

Speed is very important to perform encryption on data in real time (McLoone et. al., 2001). Different applications of the Advanced Encryption Standard (Rijndael) algorithm may require different speed / area trade-offs. Some applications, such as smart cards and cellular phones, require small area. Other applications such as WWW servers and ATMs are speed critical (Xinmiao et. al., 2002). In the computational operation in AES, several steps have to use inverse and multiplication functions. Those function include in the Mixcolumn transformation which is one of the transformation in the AES algorithm. Inverse and multiplier are the most complex modules with longer delay. Mixcolumn transformation is suggested to use look-up tables to solve the problem of delay. That function can be integrated into a bigger table to replace the calculations of inverse and multiply operations, if it provides enough memory. In fact, too many tables are needed for various irreducible polynomials so that this system is not flexible and expandable (Jing et. al., 2001). According to Xinmiao et. al. (2002), the area for lookup tables becomes huge when multiple round units are implemented. In this research we proposed the use of cyclic shift and XOR operation to replace the use of look-up tables. This approach is used to speed up Mixcolumn transformation. Detail description will be explained in chapter 4.